Take-Home Exam: Building a Retrieval-Augmented Generation (RAG) System

Submission Guidelines:

- Ensure all code is well-documented and follows best practices.
- Include a README file with instructions on how to run the code.
- Submit all deliverables in a single zip file or via a shared link to a repository.
- Prefer to use jupyter notebooks

Objective:

Design and implement a Retrieval-Augmented Generation (RAG) system using a provided dataset of example question-answer pairs.

Instructions:

- Complete each section in the order presented.
- Provide clear and concise explanations for your approach and methodology.
- Submit your code, outputs, and a brief report summarizing your findings and decisions.

Section 1: Understanding the Dataset

Task:

1. Load and explore the provided question-answer dataset.
2. Perform basic data cleaning if necessary.

Questions:

1. Describe the structure of the dataset.
2. Identify and handle any missing or inconsistent data.

Deliverables:

- A summary of the dataset.

- Code snippets used for loading and cleaning the data.
- A brief report on the data cleaning process.

Section 2: Setting Up the Retrieval System

Task:

1. Implement a retrieval system using the newest techniques in the field. You may consider one of the following advanced methods:
   - **Knowledge-Infused Generation:** This approach conditions the generation of text on retrieved vectors using techniques like knowledge attention and context encoding.
   - **Iterative Reranking:** Perform multiple retrieval and generation cycles to refine results progressively.
   - **Multi-Task Optimization:** Jointly train the retriever and generator components for enhanced coherence.
   - **Corrective Retrieval Augmented Generation (CRAG):** Enhance robustness and reliability by using an evaluator to assess the relevance of retrieved documents, triggering actions like correction, expansion via web searches, or combining internal and external knowledge.
   - **Modular RAG:** Implement a modular approach that allows for configurable components such as search, memory, and reranking modules. This includes advanced strategies like retrieval-enhanced generators, reranking models, and multi-query fusion techniques.

Questions:

1. Explain your choice of retrieval technique and the reasons behind selecting it over others.
2. Demonstrate how you retrieve the most relevant answers for a given set of questions using the chosen technique.

Deliverables:

- Code snippets for the retrieval system implementation.
- Examples of retrieval results for a few sample questions.
- An explanation of your choice and any trade-offs considered.

Section 3: Integrating the Generation Component

Task:

1. Integrate a generative model (e.g., GPT-4o) to provide more detailed and contextual answers.
2. Fine-tune the generative model if necessary.

Questions:

1. Describe the process of integrating the generative model with the retrieval system.
2. Provide examples of questions and the corresponding generated answers.

Deliverables:

- Code snippets for integrating the generative model.
- Examples of generated answers.
- A brief discussion on the challenges faced and how they were addressed.

Section 4: Evaluation and Optimization

Task:

1. Evaluate the performance of the RAG system.
2. Optimize the system based on the evaluation results.

Questions:

1. Describe the evaluation metrics used and justify your choice.
2. Present the evaluation results and discuss any optimizations performed.

Deliverables:

- Code snippets for the evaluation and optimization process.
- Evaluation results (e.g., tables, charts).

- A summary of optimizations and their impact on performance.

Section 5: Reporting and Documentation

Task:

1. Compile a final report summarizing the entire process.
2. Include explanations, code snippets, results, and any additional observations.

Deliverables:

- A comprehensive final report.
- All relevant code files and scripts.
- Additional documentation (e.g., README file, instructions for running the code).

[Extra Point] Section 6: Expanding the RAG System to the Internet

Task:

1. Extend the RAG system to retrieve information not only from the provided dataset but also from the entire internet.
2. Implement a method to evaluate the quality and reliability of the retrieved information from the web.

Questions:

1. Describe how you will expand the RAG system to include internet retrieval.
2. Explain the techniques you will use to assess and ensure the quality and reliability of the retrieved information.
3. Demonstrate how the extended RAG system retrieves and generates answers for a new set of questions.

Deliverables:

- Code snippets for integrating internet retrieval into the RAG system.

- Examples of questions, the retrieved information from the web, and the corresponding generated answers.
- A brief report on the methods used to evaluate and ensure the quality of the internet-sourced information.